Apple II: Interrupt Handling (2/97)

Revised: 3/3/97
Security: Everyone

Apple II: Interrupt Handling (2/97)

==========

Article Created: 21 September 1984
Article Reviewed/Updated: 28 February 1997

TOPIC ----------

This article describes interrupt handling routines in the Apple II.

DISCUSSION ----------

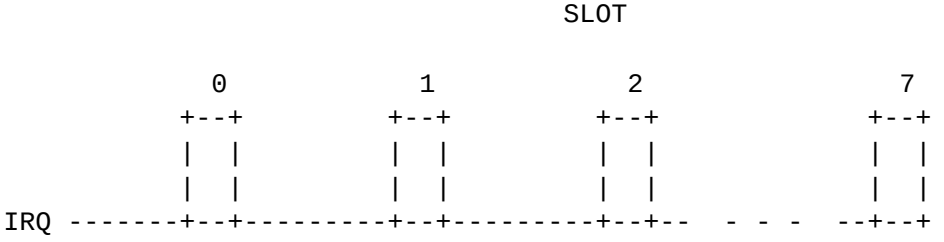
The user should be familiar with the 6502 interrupt requirements as defined in the Synertek Programming and Hardware manuals. This article applies to the interrupt request (IRQ). The use of the non-maskable interrupt (NMI) in a disk system is not recommended. The data and programs on the disk may be destroyed if an NMI occurs while the Apple is writing data to the disk. The DOS disables IRQ during critical code making it relatively safe to use.

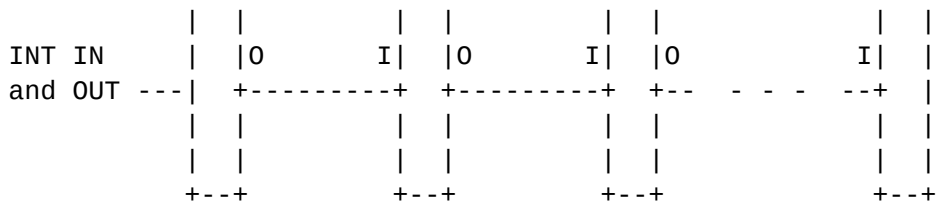
HARDWARE

For interrupts, the Apple peripheral slots have defined three pins:

IRQ, Interrupt ReQuest	pin 30
INT IN, daisy chain in	pin 28
INT OUT, daisy chain out	pin 23

The daisy chain structure allows an interface card to disable the next higher number card from requesting an interrupt. Slot 0 has the highest priority and slot 7 has the least.





The system was designed so that if INT IN for a slot is low then that slot may not generate an interrupt. Each slot should pull INT OUT low when it is requesting an interrupt to disable lower priority slots. Cards that don't use IRQ should wire INT IN and INT OUT together so that any higher priorty slots can still disable cards in lower priority slots. This priorty system fails if there is an empty slot between any two interrupting cards.

The hardware logic required to generate INT OUT is INT IN anded with the active low interrupt signal from the peripheral device. IRQ is INT IN anded with the active high interrupt signal from the peripheral device.

SOFTWARE

There are two ways to cause the 6502 to follow the IRQ vector. A logical zero on the IRQ pin of the 6502 while the IRQ flag of the processor is cleared, or executing a BREAK instruction in a program:

First, the Apple monitor determines whether a BREAK or an IRQ has occured. In the Auto start ROM, this routine is at $FA40 and in the old monitor it's at $FA86. This routine stores the 6502 accumulator at location $45 and retrieves the processor status flags. A BREAK drops into the monitor with the address of the BREAK operation code + 2 and a dump of the 6502 registers. The Auto-Start ROM has the option of jumping to a user's routine after a break. Both monitor ROMs jump to the address contained in memory at $3FE and $3FF after an IRQ.

Interrupt Request

The user must have the address of his interrupt handler stored in $3FE and $3FF before the first interrupt is generated.

Caution: The accumulator does not contain valid data when it is vectored to $3fe and $3FF.

The accumulator must be restored from location $45 before the return from interupt instruction, (RTI) is executed. The user must also be careful to leave the other registers as they were when an interrupt occured.

Interrupts and BASIC

If the user is careful to restore all the 6502 registers and not disturb BASIC's memory locations in the interrupt handling routine, the interrupt will be transparent to BASIC. Be very careful of page 0 locations. Save and restore any information on the stack when you're not sure.

Applesoft and Integer BASIC both use the 6502 stack extensively in keeping track of GOSUBs and FOR-NEXT loops. This makes it difficult to have an interrupt modify BASIC program execution. To do this the easiest way, program the interrupt routine to set a flag byte when an event occurs and then program the BASIC program to PEEK that flag byte's address and respond when the flag byte is set.

You may use the Applesoft ONERR GOTO statement to modify the execution of an Applesoft program when the interrupt occurs. The following machine language routine causes an 'error' condition in Applesoft.

```
10 POKE 800,162: POKE 801,100: POKE 802,104: POKE 803,104
20 POKE 804,104: POKE 805,76 : POKE 806,233: POKE 807,242
30 POKE 1022,0 : POKE 1023,8
40 ONERR GOTO 1000
50 PRINT "NO INTERRUPT"
60 GOTO 50
1000 IF PEEK (222) <> 100 THEN END
1010 PRINT "INTERRUPT!!!"
1020 RESUME
```

The POKEs set the IRQ vector to generate an error number 100 when an interrupt occurs. The Applesoft onerr routine can check decimal location 222 and if it doesn't equal 100 then you have a normal Applesoft or DOS error. Treat the IRQ generated error like any other Applesoft error. RESUME and the routine on page 82 of the Applesoft reference manual will work normally. Please do a search on ON ERR GOTO for more information.

Interrupts and DOS

The interrupt checking routine in the monitor saves the 6502 accumulator at location $45 while it checks for a break. Unfortunately, DOS also uses location $45 as temporary storage while DOS parses the numeric parts of its commands. This can result in range errors or reading the wrong record, slot, or drive if an interrupt occurs during parsing. There is no way around this problem at this time. To use interrupts with DOS in the safest way, disable IRQ when doing any disk access.

Article Change History:
28 Feb 1997 - Reviewed for technical accuracy, revised formatting.

Tech Info Library Article Number:107